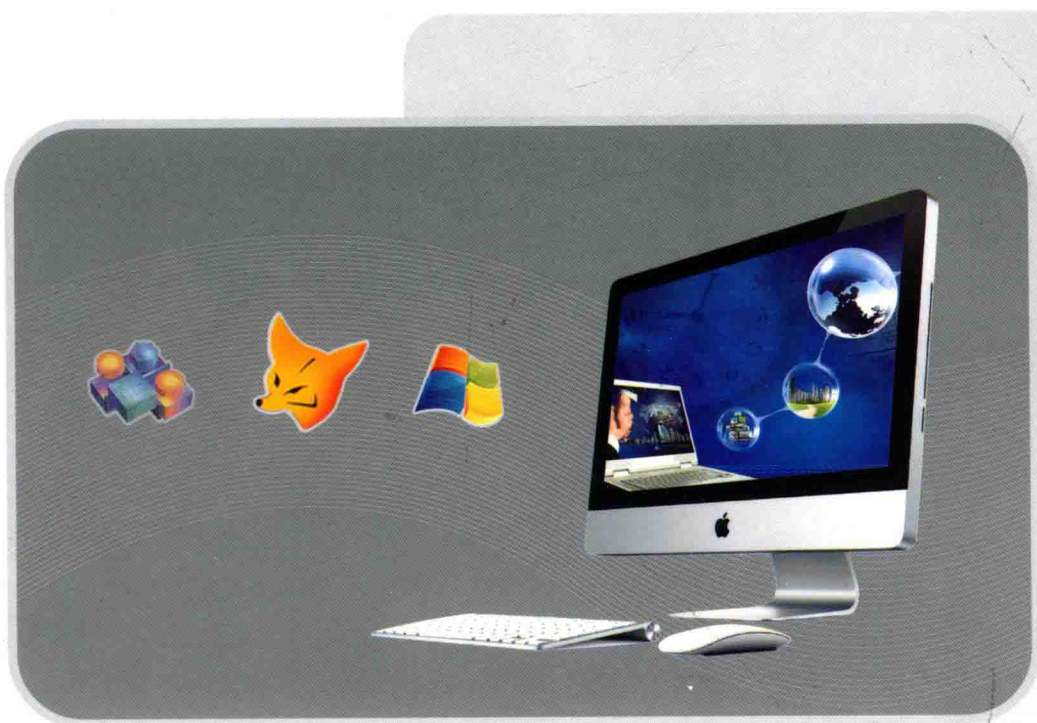




普通高等教育“十二五”重点规划教材 计算机基础教育系列

C语言实例演练

刘莹昕◎主 编



科学出版社

普通高等教育“十二五”重点规划教材 计算机基础教育系列

C 语言实例演练

刘莹昕 主编

张 宇 张春芳 姜 雪 副主编

科学出版社

北 京

内 容 简 介

本书由长期从事 C 语言教学的一线教师编写,通过大量的案例来加深读者对知识点的理解,使其能深入浅出、循序渐进地掌握 C 语言程序设计的基本思想和方法。

本书共 6 章,第 1 章介绍了 Visual C++ 6.0 的实验环境;第 2 章基础篇是根据全国计算机等级考试二级 C 语言程序设计考试大纲的知识点设计了 66 个基本实例,适用于初学者对基础知识的学习和理解;第 3 章提高篇,根据知识点的重点和难点,与实际相结合设计了 33 个实例,旨在进一步提高对知识的掌握;第 4 章综合篇是 C 语言程序设计的综合展现,让读者体会到 C 语言的乐趣和实际应用;第 5 章竞赛篇是为了让读者对程序设计大赛有更多的了解和认识,增强学习的兴趣;第 6 章是全国计算机等级考试二级 C 语言程序设计的最新考试大纲、两套试题及答案,供读者学习参考。

本书既可作为于本科、专科等高校公共课或专业基础课的教材,也可作为实验教学用书。

图书在版编目(CIP)数据

C 语言实例演练/刘莹昕主编. —北京:科学出版社,2014

(普通高等教育“十二五”重点规划教材·计算机基础教育系列)

ISBN 978-7-03-039540-5

I. ①C… II. ①刘… III. ①C 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 005076 号

责任编辑:陈晓萍 郭丽娜/责任校对:马英菊

责任印制:吕春珉/封面设计:东方人华平面设计部

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

新科印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2014 年 1 月第 一 版 开本:787×1092 1/16

2014 年 1 月第一次印刷 印张:16 1/4

字数:370 000

定价:33.00

(如有印装质量问题,我社负责调换<新科>)

销售部电话 010-62142126 编辑部电话 010-62138978-2009

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

C 语言程序设计是一门实践性很强的课程，需要大量的案例实践才能熟练掌握和应用。为了帮助初学者更好地学习 C 语言，能够熟练地使用 C 语言编写程序，我们编写了本书。

本书采用理论与实践相结合，教与学相长的编写原则，以全国计算机等级考试二级 C 语言程序设计考试大纲作为贯穿全书的总线，选用 Visual C++ 6.0 作为实验环境，概括了考试大纲中的主要内容，并针对各主要知识点，精心设计了实例。全书程序设计分为基础篇、提高篇、综合篇和竞赛篇。遵循由浅入深、循序渐进的教学原则，采用“理实一体”的教学方法，将知识点与实例相结合，初学者通过实例验证程序、修改程序、调试程序和编写程序，能够逐步领会和掌握 C 语言程序设计的实际应用和程序设计的基本思想和方法，并在举一反三环节中，将知识消化吸收，使学生达到灵活运用知识的能力，又便于教师组织课堂教学。同时，案例中引入了有趣的生活实例、小游戏和程序设计大赛的相关试题，有利于扩展学生的知识面和提高自己的学习能力。

本书由刘莹昕、张宇、张春芳、姜雪编写，刘莹昕负责全书的统稿。在本书的编写过程中，得到了刘立君、王立武、苏瑞、邵一川、王锦、朱姬凤等教师的热情帮助和支持，在此表示衷心感谢。

在本书的编写过程中，借鉴了许多现行教材的宝贵经验，在此，谨向这些作者表示诚挚的感谢。由于时间仓促及编者水平有限，本书错误和不当之处在所难免，恳请读者批评指正。

目 录

第 1 章 Visual C++ 6.0 集成开发环境	1
1.1 Visual C++ 6.0 集成开发环境的启动	1
1.2 Visual C++ 6.0 集成开发环境的使用	2
1.3 C 程序的运行步骤	7
第 2 章 基础篇	11
实例 2.1 整型数据的输出	11
实例 2.2 实型数据的输出	12
实例 2.3 实型数据的精度	14
实例 2.4 字符型数据的输出	15
实例 2.5 附加格式说明符的使用	16
实例 2.6 转义字符	18
实例 2.7 格式输入函数 scanf()	20
实例 2.8 scanf()函数中附加格式说明符的使用	22
实例 2.9 算术运算符和表达式	23
实例 2.10 自增(++)、自减(--)运算符	25
实例 2.11 赋值运算符和赋值表达式	27
实例 2.12 类型转换	28
实例 2.13 关系运算和逻辑运算	29
实例 2.14 位运算符及其表达式	32
实例 2.15 将 short 类型数据的高、低位互换	34
实例 2.16 条件运算符和条件表达式	35
实例 2.17 逗号运算符和 sizeof 运算符	36
实例 2.18 getchar()和 putchar()函数	37
实例 2.19 计算圆的周长和面积	39
实例 2.20 两个数交换	40
实例 2.21 if-else 语句	41
实例 2.22 if 多分支语句	43
实例 2.23 switch-case 语句	45
实例 2.24 while 语句	47
实例 2.25 do-while 语句	48
实例 2.26 for 循环语句	49
实例 2.27 break 和 continue 语句	51
实例 2.28 循环嵌套	52
实例 2.29 循环应用(1)	54

实例 2.30	循环应用 (2)	55
实例 2.31	一维数组	57
实例 2.32	二维数组	59
实例 2.33	字符数组	61
实例 2.34	字符串处理函数	62
实例 2.35	函数的定义	64
实例 2.36	函数的返回值	66
实例 2.37	函数的参数传递——值传递	67
实例 2.38	函数的参数传递——地址传递	69
实例 2.39	函数的嵌套调用	70
实例 2.40	函数的递归调用	72
实例 2.41	局部变量和全局变量	74
实例 2.42	静态局部变量和动态局部变量	76
实例 2.43	编译预处理——宏定义	78
实例 2.44	编译预处理——文件包含	80
实例 2.45	指针变量的引用	81
实例 2.46	指针变量作为函数	83
实例 2.47	指向一维数组的指针变量	85
实例 2.48	指向二维数组的指针变量	87
实例 2.49	指针数组	89
实例 2.50	字符串的指针	91
实例 2.51	指向指针的指针	93
实例 2.52	函数指针变量和指针型函数	94
实例 2.53	结构体变量初始化	96
实例 2.54	结构体数组	99
实例 2.55	结构体指针	101
实例 2.56	用 typedef 定义类型	103
实例 2.57	链表的建立	105
实例 2.58	链表的插入和删除	107
实例 2.59	链表的输出	110
实例 2.60	共用体存储形式	111
实例 2.61	共用体变量的赋值	114
实例 2.62	文件的字符输入和输出	116
实例 2.63	文件的字符串输入和输出	120
实例 2.64	文件的格式化输入和输出	122
实例 2.65	文件的数据块输入和输出	124
实例 2.66	文件的定位	125
举一反三	参考答案	128

第3章 提高篇	143
实例 3.1 计算本息和	143
实例 3.2 整数的逆值	144
实例 3.3 求解一元二次方程 $ax^2+bx+c=0$	144
实例 3.4 体形判断	146
实例 3.5 在屏幕上显示菜单	147
实例 3.6 水仙花数	148
实例 3.7 最大公约数与最小公倍数	150
实例 3.8 牛顿迭代法	151
实例 3.9 猴子吃桃	151
实例 3.10 百元百鸡	153
实例 3.11 韩信点兵	154
实例 3.12 三色球	155
实例 3.13 冒泡法排序	156
实例 3.14 选择法排序	157
实例 3.15 二分法查找数据	158
实例 3.16 杨辉三角形	159
实例 3.17 幻方阵问题	162
实例 3.18 凯撒加密问题	164
实例 3.19 验证哥德巴赫猜想	165
实例 3.20 使用递归方法实现正整数的逆序输出	166
实例 3.21 汉诺塔问题	168
实例 3.22 数制转换	169
实例 3.23 字符串长度	170
实例 3.24 字符串连接	171
实例 3.25 字符串的拆解	172
实例 3.26 统计选票问题	174
实例 3.27 学校人员数据管理	175
实例 3.28 创建链表	178
实例 3.29 荷兰国旗问题	179
实例 3.30 成绩排序	181
实例 3.31 文件加密/解密	185
实例 3.32 将输入字符存入文件	187
实例 3.33 将小写字母转换成大写字母并存入文件	188
第4章 综合篇	190
实例 4.1 掷骰子游戏	190
实例 4.2 大整数求和	191
实例 4.3 折半查找	193
实例 4.4 两个文件的连接	196

实例 4.5 电子英汉词典	198
实例 4.6 学生成绩管理	205
第 5 章 竞赛篇	213
试题 1 为表达式求值	213
试题 2 简单排序	215
试题 3 砵码问题	217
试题 4 评委打分问题	218
试题 5 母牛生小牛问题	219
第 6 章 全国计算机等级考试二级 C 语言程序设计试题及答案	221
6.1 全国计算机等级考试二级 C 语言程序设计考试大纲	221
6.2 全国计算机等级考试二级 C 语言试卷一	223
6.3 全国计算机等级考试二级 C 语言试卷一参考答案	233
6.4 全国计算机等级考试二级 C 语言试卷二	234
6.5 全国计算机等级考试二级 C 语言试卷二参考答案	243
附录	244
附录 A C 语言关键字	244
附录 B 常用字符与 ASCII 码对照表	245
附录 C 运算符的优先级与结合性	247
附录 D 常见错误	248
参考文献	250

第 1 章 Visual C++ 6.0 集成开发环境

Visual C++是微软公司的 Visual Studio 工具集的重要组成部分。Visual C++提供了一个集源程序编辑、代码编译和调试于一体的可视化开发环境，即所谓的集成开发环境（Integrated Development Environment, IDE）。Visual C++包含文本编辑器、资源编辑器、工程编译工具、源代码浏览器、集成调试工具，以及一套联机文档。开发环境是程序员同 Visual C++的交互界面，通过它，程序员可以完成创建、调试、修改应用程序等的各种操作。对于集成开发环境的熟悉程度直接影响到程序设计的效率。集成开发环境采用标准的多窗口 Windows 用户界面，并增加了一些新特性，使得开发环境更易于使用，用户能够很容易地学会并掌握 Visual C++的使用方法。

Visual C++ 6.0 是微软公司推出的 Visual C++开发工具的 6.0 版本，它不仅支持 Internet 特性，还增加了 ADO（ActiveX Data Objects）数据绑定、ATL（Active Template Library，活动模板库）复合控件、编辑和继续特性等，开发者可以充分利用 Visual C++ 6.0 的这些特性，构建自己的 Windows 应用程序。

由于 Visual C++是一个可视化的开发工具，在介绍 Visual C++ 6.0 的各个组成部分之前，首先要了解可视化编程的概念。可视化技术是当前发展迅速并引人注目的技术之一，它的特点是把原来抽象的数字、表格、功能逻辑等用直观的图形、图像形式表现出来。可视化编程，亦即可视化程序设计：以“所见即所得”的编程思想为原则，力图实现编程工作的可视化，即随时可以看到结果，程序与结果的调整同步。可视化编程是与传统的编程方式相比而言的，这里的“可视”，指的是无需编程，仅通过直观的操作方式即可完成界面的设计工作，是目前最好的 Windows 应用程序开发工具。这种可视化的编程方法易学、易用，而且大大提高了工作效率。在 Visual C++ 6.0 中提供了大量的用于可视化编程的资源编辑工具。

1.1 Visual C++ 6.0 集成开发环境的启动

在计算机上安装完 Visual C++ 6.0 应用程序后，可以选择以下两种方式启动。

(1) 单击 Windows “开始” 菜单，选择“程序”组下“Microsoft Visual Studio 6.0”子菜单下的“Microsoft Visual C++ 6.0”选项，启动 Visual C++ 6.0。

(2) 单击 Windows “开始” 菜单，选择“运行”命令，输入 msdev，即可启动 Visual C++ 6.0。

Visual C++ 6.0 启动后的界面如图 1.1 所示。

Visual C++ 6.0 是一个 Windows 应用程序，启动后显示一个标准的窗口，由标题栏、菜单栏、工具栏、工作区窗口、源代码编辑窗口、输出窗口和状态栏组成。

主窗口的左侧是项目工作区窗口，右侧是程序编辑区窗口，下方是输出窗口。工作区窗口显示所设定的工作区的信息，程序编辑窗口用来输入和编辑源程序，输出窗口用

来显示程序出错信息和结果有无错误（errors）或警告（warnings）。

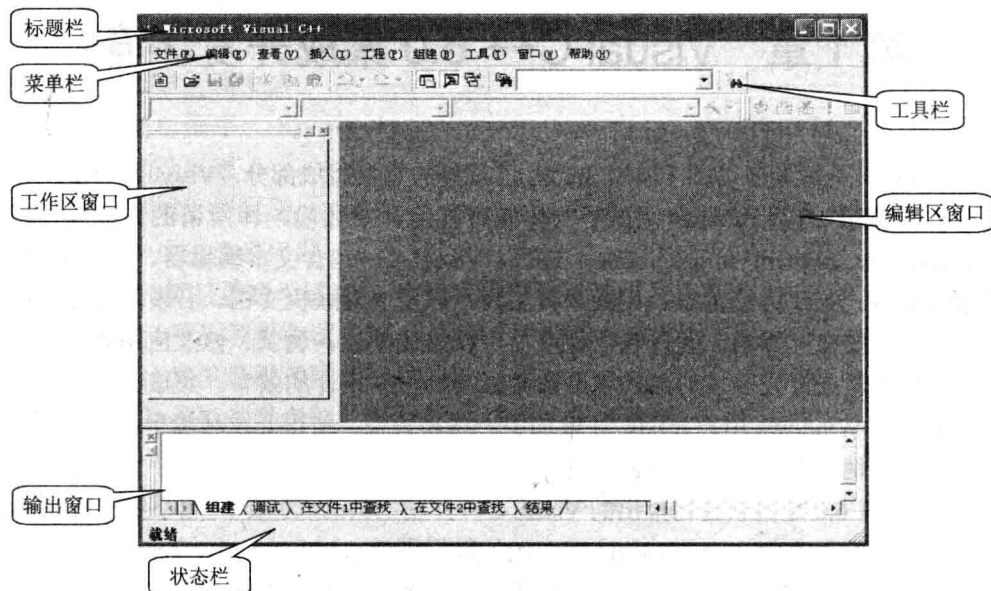


图 1.1 Visual C++ 6.0 的启动界面

1.2 Visual C++ 6.0 集成开发环境的使用

1.2.1 菜单栏

在 Visual C++ 6.0 集成开发环境中，标题栏下方是菜单栏，由九个菜单项组成。单击菜单项弹出下拉式菜单，可使用这些菜单项实现集成环境的各种功能。菜单选项及其功能介绍如下。

1) 文件

“文件”（File）菜单包括对文件、项目、工作区及文档进行文件操作的相关命令或子菜单。

2) 编辑

“编辑”（Edit）菜单不仅包括常用的剪切、复制、粘贴命令，还包括为调试程序设置的断点命令，用于实现设置、删除、查看断点。此外，还有为方便程序员输入源代码的 List Members、Type Info 等命令。

3) 查看

“查看”（View）菜单中的命令主要用于改变窗口和工具栏的显示方式、检查源代码、激活调试时所用的各个窗口等。

4) 插入

“插入”（Insert）菜单包括创建新类、新表单、新资源以及新的 Atl 对象等命令。

5) 工程

使用“工程”（Project）菜单可以创建、修改和存储正在编辑的工程文件。

6) 组建

“组建”(Build)菜单用来编译、连接程序,创建及执行应用程序。

7) 工具

“工具”(Tools)菜单允许用户简单快速地访问多个不同的开发工具,如定制工具栏与菜单、激活常用的工具(Spy++等)或者更改选项等。

8) 窗口

“窗口”(Window)菜单的命令用来进行有关窗口的操作,常用的命令有新建、分割、组合、关闭和全部关闭等。

9) 帮助

与标准“帮助”(Help)菜单相似,帮助命令用来获得大量的帮助信息。

集成开发环境还为一些常用的命令分配了默认的快捷键,记住这些常用的快捷键,用户就可以更高效地使用集成开发环境。其中常用的快捷键如表 1.1 所示。

表 1.1 常用快捷键

快捷键	功能	快捷键	功能
Ctrl+O	打开文件 (File Open)	F4	下一条错误
Ctrl+S	保存文件 (File Save)	Shift+F4	上一条错误
Ctrl+C	编辑复制 (Edit Copy)	Ctrl+F6	下一个窗口
Ctrl+X	编辑剪切 (Edit Cut)	Alt+Enter	对象属性
Ctrl+V	编辑粘贴 (Edit Paste)	Ctrl+F7	编译文件
Ctrl+F	编辑查找 (Edit Find)	F7	建立可执行文件
Ctrl+Z	编辑取消 (Edit Undo)	Ctrl+F5	运行 (Execute)

1.2.2 工具栏

菜单栏下方是工具栏,它由若干个功能按钮组成,单击某个按钮就可实现某种操作功能。工具栏也是 IDE 的重要组成部分,主要列出了在代码开发过程中经常用到的一些功能,具有直观和快捷的特点,熟练使用这些工具按钮将大大提高工作效率。在 Visual C++ 6.0 开发环境中包括 10 个标准工具栏,这些工具栏并不都显示在开发环境中,可以在工具栏上任意位置右击,在弹出的快捷菜单中选择要显示的工具栏,如图 1.2 所示。

下面介绍三个常用的工具栏。

1. 标准工具栏

标准 (Standard) 工具栏用于维护和编辑工作区的文本和文件,如图 1.3 所示。

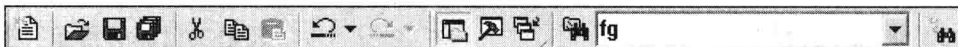
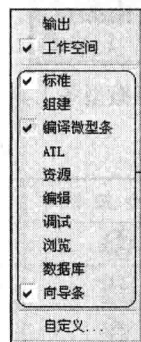


图 1.3 标准工具栏



工具栏

图 1.2 工具栏快捷菜单

标准工具栏中各命令按钮的功能说明如表 1.2 所示。

表 1.2 标准工具栏中各命令按钮的功能说明

命 令 按 钮	功 能 说 明
	创建一个新的文件、项目和工作区
	打开一个已存在的文件、项目和工作区
	保存当前打开的文件
	保存所有打开的文件
	将所选择的内容剪切掉，移到剪贴板中
	将所选内容复制到剪贴板中
	在当前位置粘贴剪贴板中最新的内容
	取消上一次的操作
	恢复被取消的操作
	激活工作区窗口，用来管理工程中的文件和资源
	激活输出窗口，用来显示编译、调试和查找的信息
	管理当前打开的窗口
	在所有窗口中查找指定字符串
	在当前窗口中查找指定字符串
	调用 MSDN

2. 编译微型条工具栏

编译微型条 (Build MiniBar) 工具栏用于运行程序和调试程序，如图 1.4 所示。



编译微型条工具栏中各命令按钮的功能说明如表 1.3 所示。

图 1.4 编译微型条工具栏

表 1.3 编译微型条工具栏中各命令按钮的功能说明

命 令 按 钮	功 能 说 明
	用于编译当前在源代码编辑窗口的源文件
	用于编译、连接当前工程中的文件，生成一个可执行文件
	终止编译或连接的程序
	运行程序
	开始或继续调试程序
	编辑程序中的断点

3. 调试工具栏

调试 (Debug) 工具栏用于调试程序，如图 1.5 所示。

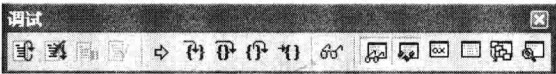


图 1.5 调试工具栏

调试工具栏中各命令按钮的功能说明如表 1.4 所示。

表 1.4 调试工具栏中各命令按钮的功能说明

命 令 按 钮	功 能 说 明
	放弃当前的调试，重新开始调试
	终止调试，返回到编辑状态
	暂停调试
	改编代码后调试
	显示将要运行的代码行
	单步执行程序，进入到函数内部
	单步执行程序，不进入到函数内部
	跳出当前函数
	运行到光标
	弹出 Quick Watch 窗口
	显示/隐藏 Watch 窗口
	显示/隐藏 Variables 窗口
	显示/隐藏 Registers 窗口
	显示/隐藏 Memory 窗口
	显示/隐藏 Call Stack 窗口
	显示/隐藏 Disassembly 窗口

在如图 1.2 所示工具栏快捷菜单中可以选择“自定义”命令，弹出“定制”对话框，如图 1.6 所示。用户可以根据实际需求定制工具栏。

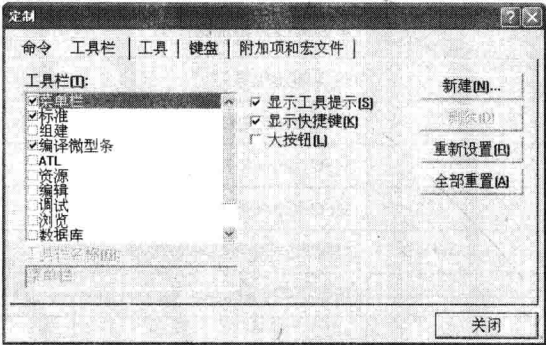


图 1.6 “定制”对话框

1.2.3 项目和项目工作区

Visual C++应用程序的核心是工程（Project，也称“项目”），通常位于工程工作区

(Workspace) 中, Visual C++ 工程工作区可以容纳多个项目。

1. 工程

在 Visual C++ 6.0 集成开发环境中, 可以通过选择“文件→新建”菜单命令, 通过“新建”对话框来创建一个新的工程。这个工程可以是一个游戏程序, 或者是其他的应用程序。工程名是工程中其他文件命名的基础, 其扩展名为.dsp (Developer Studio Project)。

“新建”对话框的“工程”选项卡内容如图 1.7 所示。

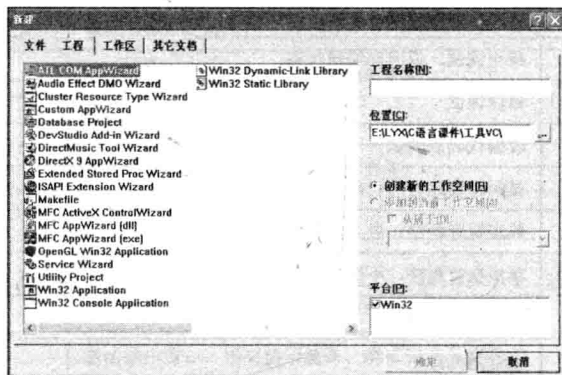


图 1.7 “新建”对话框中“工程”选项卡

“工程”选项卡中可供选择的工程类型名称及含义如表 1.5 所示。

表 1.5 工程类型名称及含义

类 型	含 义
ATL COM AppWizard	ATL 应用程序创建向导
Cluster Resource Type Wizard	簇资源类型创建向导
Custom AppWizard	自定义的应用程序创建向导
DataBase Project	数据库工程
DevStudio Add-in Wizard	插件创建向导
Extended Stored Proc Wizard	扩展存储编程创建向导
ISAPI Extension Wizard	扩展创建向导
Makefile	C/C++生成文件
MFC ActiveX ContorWizard	MFC ActiveX 控制程序创建向导
MFC AppWizard (exe)	MFC 可执行程序创建向导
MFC AppWizard (dll)	MFC 动态链接库创建向导
New DataBase Wizard	新数据库创建向导
Utility Project	单元工程
Win32 Applicator	Win32 应用程序
Win32 Console Application	Win32 控制台应用程序
Win32 Dynamic-Link Library	Win32 动态链接库
Win32 Static Library	Win32 静态库

需要注意的是,要建立一个扩展名是.c 的程序文件,在“工程”选项卡中要选择“Win32 Console Application”。“Win32 Console Application”的入口函数是 main()函数。

2. 工程工作区

在创建一个工程的同时,也创建了一个工程工作区。工程工作区文件的扩展名为.dsw (Developer Studio Workspace)。通过工程工作区窗口可以查看和访问工程的各种组件,主要用于保存工作区的设置。工程工作区文件含有工作区的定义和工程中所包含文件的全部信息。

在创建一个工程的同时, Visual C++ 6.0 为这个工程创建了一个默认的工程工作区。

1.3 C 程序的运行步骤

1.3.1 编辑

启动 Visual C++ 6.0 以后,选择“文件”菜单下的“新建”菜单项,出现如图 1.8 所示的“新建”对话框。选择“文件”选项卡,选择“C++ Source File”文件类型,输入程序名(程序的扩展名是.c,否则系统默认为 C++源程序的扩展名是.cpp)及保存程序的目录。

注意:输入文件名时要指定的文件名扩展名为.c,如果输入的文件名为*.cpp,则表示要建立的是 C++源程序。如果不写扩展名,系统会默认指定为 C++源程序文件,自动加上扩展名.cpp,因此,编写 C 语言程序不能省略扩展名.c。

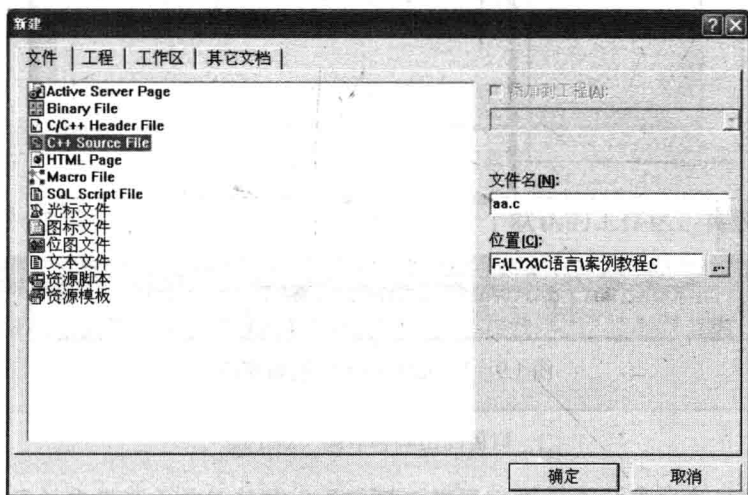


图 1.8 “新建”对话框中的“文件”选项卡

“文件”选项卡中显示出可创建的文件类型名称及含义,如表 1.6 所示。

单击“确定”按钮,出现如图 1.9 所示的 Visual C++ 6.0 编辑窗口,在此窗口中输入 C 程序代码。

表 1.6 文件类型名称及含义

类 型	含 义
Active Server Page	服务器页文件
Binary File	二进制文件
Bitmap File	位图文件
C/C++ Header File	C/C++头文件
C++Source File	C++源程序文件
Cursor File	光标文件
HTML Page	HTML 页文件
Icon File	图标文件
Macro File	宏文件
Resource Script	资源脚本文件
Resource Template	资源模板文件
SQL Script File	SQL 脚本文件
Text File	文本文件



图 1.9 Visual C++ 6.0 编辑窗口

1.3.2 编译

在 Visual C++ 6.0 环境中, 选择“组建”菜单下的“编译”菜单项, 如图 1.10 所示。对 C 语言的源程序进行编译, 并生成扩展名为.obj 的目标文件。

编译时, 在屏幕下方的调试信息窗口中将显示出编译的结果, 指出有多少错误 (error(s)) 和警告性的错误 (warning(s))。可以根据显示的错误信息所在的行号尽快地找到错误语句, 修改程序, 然后再编译。

对程序进行“编译”还可以使用系统提供的“编译微型条”工具栏。在系统提供

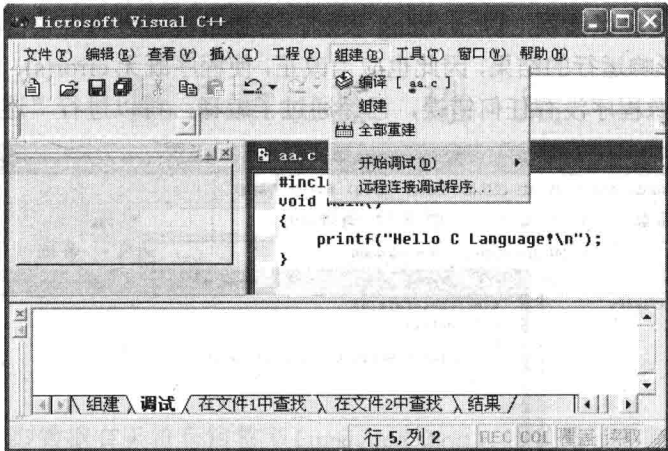


图 1.10 “组建”菜单

的“编译微型条”工具栏上，包括经常使用的“编译”、“连接”、“执行”等功能按钮，如图 1.11 所示。

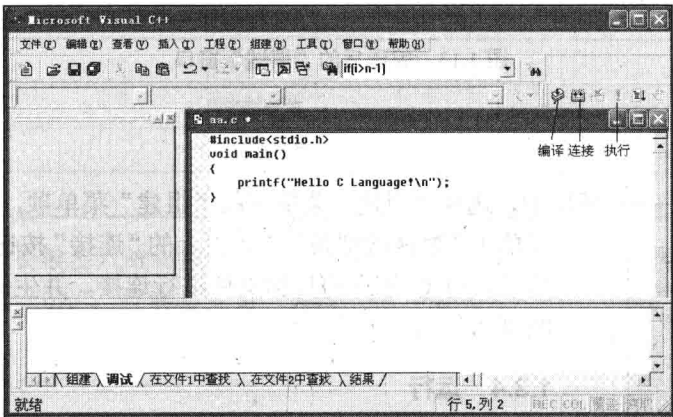


图 1.11 “编译微型条”工具栏

对程序第一次编译会弹出一个窗口，要求创建一个默认的工作区，如图 1.12 所示。

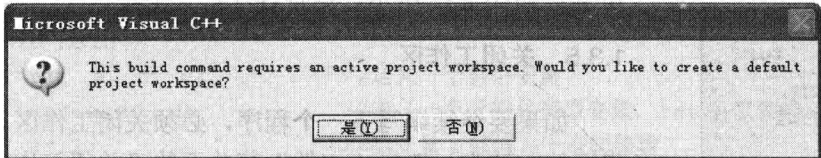


图 1.12 编译时弹出的窗口

说明：如果创建程序之前已经建立了项目工作区，则不会出现这个对话框。

单击“是”按钮，表示同意由系统建立默认的项目工作区，对如图 1.9 所示窗口中的程序进行编译后，在屏幕下方的调试信息窗口中将产生编译结果，如图 1.13 所示。

屏幕下面的调试信息窗口指出源程序有无错误，在此开始程序的调试，发现和改正程序中的错误，编译系统能检查程序中的语法错误。语法错误分为两类：一类是致命错误，以 error 表示，如果程序有这类错误，就通不过编译，无法形成目标程序，更谈不上